1 Assignment 1 (100 points)

In this assignment, you will build on the ideas from Lab 1 to build a very basic red-object detector. Due to the Monday holiday, this assignment will be due at midnight on Wednesday, January 22. To submit your work, put answers to the written questions in a file(s) at the top level directory. Clearly label all of your work. For the math-oriented portion of the assignment, you may typeset in LaTeX or Word. If you don't know how to correctly typeset math equations, write your solution very neatly, with good labels, on a piece of paper and take a clear, high resolution photograph of the paper. Work that is not legible will not be graded. Clean your Visual Studio Solution, removing all temporary files, then zip up your code, output files, and written answers. Name the file `CS585_Assignment1_username.zip`. Submit the code with web-submit.

1.1 Learning Objectives

1. Become familiar with using Visual Studio
2. Become familiar with basic image representation in OpenCV
3. Explore representation of color
4. Explore issues that arise with object detection via thresholding

1.2 Technical Task (55 points)

Using ideas that we covered in Lab 1, and the provided skeleton code,write a body for the `thresholdChannel` function, which starts on line 17 of Assignment1.cpp. Starting with the provided error checking, you should write a function body that creates an image mask delineating the pixels in the image that belong to the red object. For any pixel in the source image where the red value exceeds the threshold, the corresponding pixel in the mask image should be set to 255, otherwise it should be set to 0.

The skeleton code provided accepts two command line arguments: the name of the file and the threshold to use. You should set the command line arguments via the Properties pane. Run your code using the file "redEgg.jpg" found in the Assignment1 directory. Start with a threshold of 224. When the program is run, it will produce two windows: One showing the original image, and another showing the image mask. Continue to adjust the threshold via the command line arguments until you are happy with the result. The program will create the output file "redThreshold.png." Once you are happy with the result, create a copy of this image titled "redEgg_thresholded_[thresholdValue].png." For example, if you use a threshold of 224, your filename should be "redEgg_thresholded_224.png."

1.3 Questions (25 points)

1. (1 pt) List the decimal RGB color codes for the following colors: Red, Yellow, White.

2. (1 pt) Look at the picture "yellowEgg.jpg. What do you predict will happen when you use your red-object detector on the image of the yellow object?

3. (5 pts) Run your thresholding program on the file "yellowEgg.jpg." Create a copy of the results image entitled yellowEgg_thresholded_[thresholdValue].png. Did the result match your expectations? If not, why?

4. (5 pts) Read section 2.3.2 in the book about color, especially the section about "Other color spaces" and/or read the Wikipedia articles on the RGB, YCbCr, and HSV color spaces. (Spoiler alert!) How could you adjust your red-object detector so that it will only detect red objects and not yellow objects?

5. (5 pts) Take a picture of a red object in your environment and run your red-object detector. What difficulties come up when you work with natural images? Submit the original image and your output.

6. (1 pt) What did you find most interesting about color representation?

7. (1 pt) Calculate the number of pixels in an image that is 960 pixels wide and 720 pixels tall, then calculate the number of elements that are in a three-channel color image of the same size.

8. (1 pts) Calculate the number of additions and multiplications used to calculate image indexes in version 2 and version 3 of the `addOne_LinearIndex` function from Part 2 of the lab. Why was version 3 so much faster than version 2?

9. (5 pts) In part 2 of the assignment, we wrote several versions of a function to add one to all elements of an image. In a loop, we added one and displayed the resulting image 255 times. You may have noticed something strange happening to the colors in the image. Using what you know about integer overflow and RGB color representation, explain what is happening in the image.

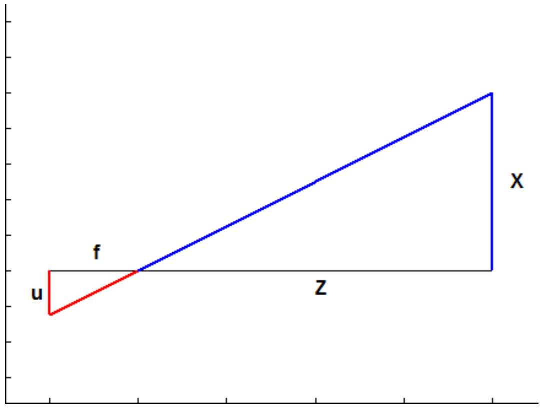
2 Lecture preparation (20 points)

2.1 Preparation for Lecture 2

In Lecture 2, we will be discussing image formation, light, and pinhole cameras. Geometry and Linear algebra are very important in computer vision, so here are a few warm-up exercises.

1. (1 pt) Write a formula for the area, A, of the base of a cone, given an apex angle α and height h

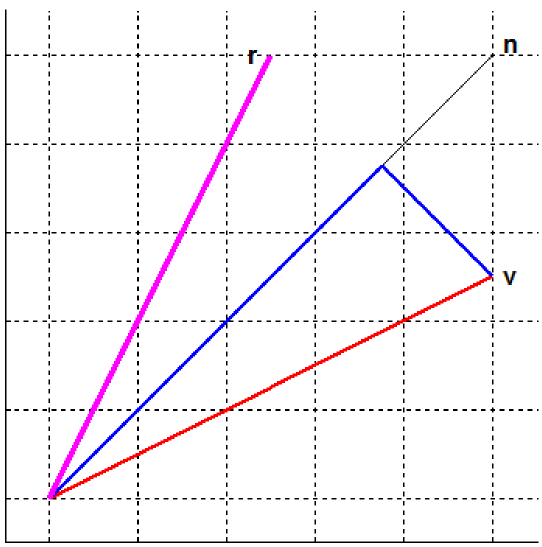
2. (1 pt) If the blue and red triangles in the figure below are similar, if $X = 5$ meters (m), $f = 25$ millimeters (mm) and $u = 540$ micrometers (μm), what is the distance Z?



3. (1 pt) If the blue and red triangles in the figure above are similar, if $X = 1$ meters (m), $f = 25$ millimeters (mm) and $Z = 10$ meters (μm), what is the distance u?

4. (1 pt) Using the numbers above, if u is the size of the image of an object on an image sensor, and your pixels have a size of 18 μm, what is the size of u in pixel units?

5. (1 pt) Look at the second figure below. Given vectors $n = [-1, 1]$ (black) and $v = [1, 0.5]$ (red), write the magnitude (length) $|n|$ and $|v|$ of the vectors and write the corresponding unit vectors $||n||$ and $||v||$.



6. (1 pt) Compute the dot product of $||n||$ and $||v||$
7. (1 pt) Compute the angle between n and v (specify degrees or radians in your answer).
8. (2 pt) Decompose the vector v into a sum of two vectors where one vector is parallel to n and the other is perpendicular to n (blue vectors). Hint: To compute the appropriate vector that is parallel to n, compute the projection of v onto $||n||$.
9. (2 pt) Compute the vector r (magenta), which is v reflected across n. (As if r were the direction of a ball bouncing off of a wall with normal direction n.)

2.2 Preparation for Lecture 3

In Lecture 3, we will be discussing image regions i.e. connected components and their properties.

1. (1 pt) Write pseudo-code for a depth-first traversal of a graph.
2. (1 pt) Given a list of all vertices in a graph, write pseudo-code for identifying the connected components of a graph.
3. (1 pt) Run your program from section 1.2 on the picture "twoRedEggs.jpg" and save the results as "twoRedEggs_thresholded_[thresholdValue].png"
4. (2 pts) Propose a strategy for counting the number of red objects in an image.
5. (2 pts) Propose a strategy for discarding the small extra regions that may pass through your red-object detector. What assumptions are you making when you propose this strategy?
6. (2 pts) Read this entertaining description of the myth of Theseus and the Minotaur (`http://edc.tversu.ru/elib/inf/0080/LiB0061.html`)and the strategy for finding the way out of a maze (you can stop about half-way through). Propose a strategy for finding an ordered list of pixels that comprise the perimeter of an image region.